Simulation of Helicopter Ditching using Smoothed Particle Hydrodynamics

Mark A. Woodgate * George N. Barakos †
CFD Laboratory, School of Engineering, James Watt South Building, University of Glasgow, Glasgow, G12 8QQ.

Nigel Scrase ‡ Tim Neville §
Leonardo Helicopters, Yeovil, BA20 2YB, UK

This paper explores the potential use of smoothed particle hydrodynamics methods for helicopter ditching. The method appears suitable for this task since it is mesh-free and can accommodate the interaction between a floating object and the free-surface of water. Simple cases of objects dropped on water were first studied to establish confidence on the method, and quantify the effect of the numerical parameters of SPH including the boundary condition between the water and solid, the effect of the number and type of smoothed particles as well as the generation of different sea-states for the ditching. Once confidence on the method was established, experiments for the ditching of a model-scale helicopter were used for validation. The smoothed particle hydrodynamics method provides good agreement with experiential data for the position and velocity of the helicopter fuselage.

Introduction

Ditching is an emergency surfacing on water, deliberately executed, with the intent of abandoning the helicopter as soon as practical. After ditching, the helicopter either floats upright, floats inverted or sinks inverted[1–3]. Between 2000 and 2003 the CAST project[4] (Crashworthiness of Helicopter on Water: Design of Structures using Advanced Simulation Tools [5]) assessed methods that could simulate ditching, with Smoothed Particle Hydrodynamics (SPH) being one of these. This work was continued in the GARTEUR HC/AG-15[6]: Improvements in SPH methods for application to helicopter ditching [7, 8] and the follow on program Smart Aircraft in Emergency Situations (SMAES)[9] between 2011 and 2014. These included looking into adding air entrapment, cavitation and suction force effects to improve both analytical [10, 11] and numerical models [12].

Water impact was first studied by von Karman [13] in the late twenties who developed a theoretical formula for water impact and compared it to experimental data from sea plane floats. The problem was idealised to the calculation of forces generated during a vertical impact of a wedge shape onto water in two dimensions. Several years later, Wagner

*Research associate, CFD Laboratory, School of Engineering, James Watt South Building, University of Glasgow, Glasgow, G12 8QQ.
†Professor, CFD Laboratory, School of Engineering, James Watt South Building, University of Glasgow, Glasgow, G12 8QQ.
‡UIC Products, Military Qualification SME, Leonardo Helicopters, Yeovil, BA20 2YB, UK.
§Principal Engineer, ASI Shipboard Operations, Leonardo Helicopters, Yeovil, BA20 2YB, UK.

[14, 15] increased the fidelity of the model by taking into account the free water surface in the form of the local liquid uprise. The Wagner model was then extended into axisymmetric cases by Chuang [16]. More recently, Scolan and Korobkin [17, 18] looked at the energy distribution from the vertical impact of 3D objects on calm water.

An assessment of two models proposed by Korobkin [19], the simplified generalised Wagner model, and the modified Logvinovich model [20, 21] which linearises the velocity potential on the wetted surface about a different point, and the model of Zhoa and Faltinsen [22] has been carried out by Tassin *et. al.* [23]. In all these analytical models, the body is assumed to be rigid and the fluid's inertia dominates the forces acting on it during the impact. The effects of viscosity, surface tension, compressibility, gravity are neglected. The flow is also assumed to be irrotational.

Regardless of the promising results obtained with these models, the need to study the impact of complex shapes on water requires a different approach that can accommodate changes of the geometry as well as multiple surfaces impacting the water at the same time. For this reason most of the modern efforts are directed towards Computational Fluid Dynamics methods that offer a general framework for ditching studies even if their computational cost is considerably higher.

In this work we compare simulated ditching of a scaled AW159 aircraft accounting for the lift of the main rotor and compare with drop test data provided by Leonardo Helicopters[24].

Numerical Method

Smoothed Particle Hydrodynamics Overview

SPH is a mesh-free method originally formulated by Lucy [25], and Gingold and Monaghan [26] that solves a set of partial differential equations both accurately and stably without using any mesh connecting the particles. SPH is an interpolation method which approximates values and derivatives of continuous variables using a set of discrete sample points. These points are smoothed particles which have a position, velocity and mass, and are calculated as some weighted average from all adjacent particles. This work builds on the SPHysics/DualSPHysics solver [27, 28] that has been extended to include a rotor model since when ditching the helicopter can still have substantial lift from the main rotor.

SPH is a computational fluid dynamics method but takes a different approach to mainstream mesh-based methods like the Helicopter Multi-Block solver of Glasgow [29, 30]. In mesh-based methods, the continuum domain is divided into discrete small sub-domains called cells. The edges of these cells form a lattice which connects the mesh points together. The governing equations are then discretised over these cells. Although mesh-based methods have been very successful they are not well suited for all types of problems. The difficulties occur when trying to keep the mesh compatible with the physical continuum and hence problems with free surfaces, deformable boundaries or moving interfaces present complications for mesh-based schemes.

The outline of the basic SPH method is shown in figure 1. The fluid is treated as a set of particles each of which has physical properties associated to them like mass, density, position and velocity. Next, a neighbour list is constructed to find the adjacent particles. This is done by cutting the computational domain into boxes of size $2h$. A list is then built of all the particles inside that box. For any given particle, only the interaction between itself and adjacent particles closer than $2h$ are to be considered so a particle can only interact with particles in the same or adjacent boxes. All particles in these boxes (9 in 2D as shown in figure 1 or 27 in 3D) are checked to find the ones within $2h$. The particle interactions can now be calculated and these forces can be used to update the physical properties of each particle.

As stated above, the SPH is an interpolation method based on the theory of integral interpolants using kernels that approximate a delta function. The integral interpolant reads:

$$f(\mathbf{x}) = \int_{\Omega} f(\mathbf{x}')\delta(\mathbf{x} - \mathbf{x}')d\mathbf{x}' \tag{1}$$

where f is a function of the three dimensional position vector $\mathbf{x}$, $\delta(\mathbf{x} - \mathbf{x}')$ is the Dirac delta function and Ω is the volume of the integral containing the point $\mathbf{x}$. If the Dirac delta function is replaced by a smoothing function $W(\mathbf{x} - \mathbf{x}')$ with the width h, then equation 1 becomes:

$$< f(\mathbf{x}) >= \int_{\Omega} f(\mathbf{x}')W(\mathbf{x} - \mathbf{x}')d\mathbf{x}' \tag{2}$$

The width h is a scaling factor that controls the smoothness/roughness of the kernel while using <>, that is the standard SPH convention. The smoothing function W is normally an even function and satisfies the following conditions. Firstly, the integration of the smoothing function W must be normalised to unity:

$$\int_{\Omega} W(\mathbf{x} - \mathbf{x}')d\mathbf{x}' = 1. \tag{3}$$

Secondly, in the limit as $h \to 0$ it must equal the Dirac delta:

$$\lim_{h \to 0} W(\mathbf{x} - \mathbf{x}') = \delta(\mathbf{x} - \mathbf{x}'), \tag{4}$$

and lastly W should be compact:

$$W(\mathbf{x} - \mathbf{x}') = 0 \quad |\mathbf{x} - \mathbf{x}'| > \kappa h \tag{5}$$

for some constant κ. This implies that only particles close to the point $\mathbf{x}$ are used in the average.

The approximation of the gradient of f is obtained by replacing $f(\mathbf{x})$ with $\nabla \cdot f(\mathbf{x})$ in equation 2

$$\begin{aligned} < \nabla \cdot f(\mathbf{x}) > &= \int_\Omega [\nabla \cdot f(\mathbf{x}')] W(\mathbf{x} - \mathbf{x}') d\mathbf{x}' \\ &= \int_\Omega \nabla \cdot [f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}')] d\mathbf{x}' - \int_\Omega f(\mathbf{x}') \cdot \nabla W(\mathbf{x} - \mathbf{x}') d\mathbf{x}' \\ &= \int_S f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}') \cdot \mathbf{n} dS - \int_\Omega f(\mathbf{x}') \cdot \nabla W(\mathbf{x} - \mathbf{x}') d\mathbf{x}' \end{aligned} \tag{6}$$

using the divergence theorem where S is the surface of the domain of integration Ω. If Ω lies within the problem domain, and since the function W has compact support the surface integral is zero. However, if Ω overlaps the problem domain, for example close to the fluid body boundary, the function W is truncated and so non zero.

If the infinitesimal volume $d\mathbf{x}'$ is replaced with the volume of the particle ΔV_j that has a corresponding mass m_j then:

$$m_j = \Delta V_j \rho_j \tag{7}$$

for each of the N particles in the support domain Ω then the numerical approximation to equation 2 is:

$$\begin{aligned} < f(\mathbf{x}) > &= \int_\Omega f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}') d\mathbf{x}' \\ &\approx \sum_j^N f(\mathbf{x}_j) W(\mathbf{x} - \mathbf{x}') \Delta V_j \\ &= \sum_j^N \frac{m_j}{\rho_j} f(\mathbf{x}_j) W(\mathbf{x} - \mathbf{x}'). \end{aligned} \tag{8}$$

The effectiveness of the SPH method depends on the choice of the weighting function. Kernels are expressed as a function of a non dimensional distance between particles given by $q = r/h$ where r is the distance between particles, and h controls the number of particles that the interactions are calculated over. There are a huge number of possible functions and some of the more common are outlined below and are shown in figure 2. Examples include the Gaussian:

$$W(r, h) = \alpha_d \exp(-q^2), \tag{9}$$

the Quadratic[31]:

$$W(r, h) = \alpha_d \left[\frac{3}{16} q^2 - \frac{3}{4} q + \frac{3}{4} \right] \quad 0 \le q \le 2, \tag{10}$$

the Cubic spline[32]:

$$W(r, h) = \alpha_d \begin{cases} 1 - \frac{3}{2} q^2 + \frac{3}{4} q^3 & 0 \le q \le 1 \\ \frac{1}{4} (2 - q)^3 & 1 \le q \le 2 \\ 0 & q \ge 2 \end{cases}, \tag{11}$$

and the Quintic[33]:

$$W(r,h) = \alpha_d \left[1 - \frac{q}{2}\right]^4 (2q+1) \quad 0 \le q \le 2. \tag{12}$$

where $\alpha_d = 1/(\pi h^2)$, $2/(\pi h^2)$, $10/(7\pi h^2)$, and $7/(4\pi h^2)$ respectively in two dimensions and $\alpha_d = 1/(\pi^{3/2} h^3)$, $5/(4\pi h^3)$, $1/(\pi h^3)$, and $21/(16\pi h^3)$ in three dimensions.

SPH for the Navier-Stokes Equations

The continuity equation in Lagrangian form is written as:

$$\frac{D\rho}{Dt} = -\rho \nabla \cdot \mathbf{v}\mathbf{v}. \tag{13}$$

There are two commonly used SPH continuity formulations used in computations derived by applying different approximation rules. Considering equation 13, one can write:

$$\begin{aligned} \left\langle \frac{D\rho}{Dt} \right\rangle &= -\langle \rho \nabla \cdot \mathbf{v} \rangle \\ &\approx -\langle \rho \rangle \langle \nabla \cdot \mathbf{v} \rangle \\ &\approx -\langle \rho \rangle \nabla \cdot \langle \mathbf{v} \rangle \end{aligned} \tag{14}$$

with

$$(\nabla . \langle v \rangle)_i = \sum_j \Delta V_j v_j \cdot \nabla_i W_{ij} = \sum_j \frac{m_j}{\rho_j} v_j \cdot \nabla_i W_{ij} \tag{15}$$

and

$$\langle \nabla W \rangle_i = \sum_j \frac{m_j}{\rho_j} \nabla_i W_{ij}. \tag{16}$$

Substituting equations (15) and (16) into equation (14) gives:

$$\frac{D\rho_i}{Dt} = \rho_i \sum_j \frac{m_j}{\rho_j} (v_i - v_j) \cdot \nabla_i W_{ij} = \rho_i \sum_j \frac{m_j}{\rho_j} v_{ij} \cdot \nabla_i W_{ij}. \tag{17}$$

The second continuity equation can be derived by applying the approximation rule for the dot product as follows

$$\langle \rho \nabla \cdot v \rangle_i \approx \sum_j (v_j - v_i) \cdot \nabla_i W_{ij} m_j \tag{18}$$

and hence

$$\frac{D\rho_i}{Dt} = \rho_i \sum_j \frac{m_j}{\rho_i}(v_i - v_j) \cdot \nabla_i W_{ij} = \rho_i \sum_j \frac{m_j}{\rho_i} v_{ij} \cdot \nabla_i W_{ij}. \tag{19}$$

It is easy to spot the difference between equation (17), that represents the summation density approximation, and (19) that represents the continuity density approximation.

The momentum equation in a continuum field, with no body force, is

$$\frac{Dv^\alpha}{Dt} = \frac{1}{\rho}\frac{\partial \sigma^{\alpha\beta}}{\partial x^\beta} \tag{20}$$

where σ is the total stress tensor made up of two parts, the isotropic pressure p and the viscous stress τ

$$\sigma^{\alpha\beta} = -p\delta^{\alpha\beta} + \tau^{\alpha\beta}. \tag{21}$$

For Newtonian fluids the viscous shear stress should be proportional to the shear strain rate via the dynamic viscosity μ, and consequently

$$\tau^{\alpha\beta} = \mu\left(\frac{\partial v^\beta}{\partial x^\alpha} + \frac{\partial v^\alpha}{\partial x^\beta} - \frac{2}{3}(\nabla \cdot v)\delta^{\alpha\beta}\right). \tag{22}$$

Below are the two most common ways in the literature to approximate the momentum equation. First consider the equation

$$\left\langle \rho \frac{Dv}{Dt} \right\rangle_i = \langle \nabla \cdot \sigma \rangle_i, \tag{23}$$

as

$$\rho_i \frac{Dv_i}{Dt} \approx \sum_j \left(\sigma_i + \sigma_j\right) \cdot \nabla_i W_{ij} \Delta V_j \tag{24}$$

we have:

$$m_i \frac{Dv_i}{Dt} = \sum_j \Delta V_i \Delta V_j \left(\sigma_i + \sigma_j\right) \cdot \nabla_i W_{ij} = \sum_j \frac{m_i m_j}{\rho_i \rho_j} \left(\sigma_i + \sigma_j\right) \cdot \nabla_i W_{ij}. \tag{25}$$

Secondly, using a different SPH gradient approximation

$$\left\langle \frac{Dv}{Dt} \right\rangle_i = \left\langle \frac{1}{\rho} \nabla \cdot \sigma \right\rangle_i, \tag{26}$$

and as

$$\frac{Dv_i}{Dt} \approx \sum_j \left(\frac{\sigma_i}{\rho_i^2} + \frac{\sigma_j}{\rho_j^2}\right) \cdot \nabla_i W_{ij} m_j \tag{27}$$

then

$$m_i \frac{Dv_i}{Dt} = \sum_j m_i m_j \left(\frac{\sigma_i}{\rho_i^2} + \frac{\sigma_j}{\rho_j^2}\right) \cdot \nabla_i W_{ij}. \tag{28}$$

Both equations (25) and (28) are symmetric with respect to the indices i and j which reduces the errors arising from the particle inconsistency problem, see for example Monaghan [34–36].

Due to its simplicity, the artificial viscosity outlined in Monaghan [37] is normally used. It provides the correct amount of viscosity to convert kinetic energy into heat at shocks and also helps to prevent unphysical penetration when two particles become close. When this term is added to the momentum equation (28) the following equation is obtained:

$$\frac{Dv_i}{Dt} = -\sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} + \Pi_{ij} \right) \cdot \nabla_i W_{ij} \tag{29}$$

where Π_{ij} is given by

$$\Pi_{ij} = \begin{cases} \dfrac{-\alpha \bar{c}_{ij} \mu_{ij} + \beta \mu_{ij}^2}{\bar{\rho}_{ij}} & v_{ij} \cdot r_{ij} < 0 \\ 0 & v_{ij} \cdot r_{ij} \geq 0 \end{cases}, \tag{30}$$

$v_{ij} = v_i - v_j$, $r_{ij} = r_i - r_j$, and

$$\mu_{ij} = \frac{h v_{ij} \cdot r_{ij}}{|r_{ij}|^2 + o^2}, \quad \bar{c}_{ij} = \frac{1}{2}(c_i + c_j), \quad \bar{\rho}_{ij} = \frac{1}{2}(\rho_i + \rho_j).$$

The expression Π_{ij} contains a linear difference in the velocity which produces both a shear and bulk viscosity. The quadratic term is required to handle high Mach number shocks, and hence β will be always set to zero. This viscosity has a number of good features. Firstly, it is invariant in Galilean transformations, secondly it conserves total linear and angular momentum, and finally it vanishes for rigid body rotations. The parameter o is to prevent the denominator going to zero and is taken so that $o = 0.1h$.

The Newtonian viscous stresses in the momentum equation can be formulated as a hybrid of a standard SPH first derivative with a finite difference approximation for the first derivative:

$$(\nu \nabla^2 v)_i = \nu \sum_j \frac{4 m_j r_{ij} \cdot \nabla_i W_{ij}}{(\rho_i + \rho_j)(|r_{ij}|^2 + \nu^2)} u_{ij} \tag{31}$$

where ν is the kinetic viscosity. The final momentum equation is:

$$\frac{Dv_i}{Dt} = -\sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \cdot \nabla_i W_{ij} + \nu \sum_j \frac{4 m_j r_{ij} \cdot \nabla_i W_{ij}}{(\rho_i + \rho_j)(|r_{ij}|^2 + \nu^2)} u_{ij}. \tag{32}$$

By contrasting equations (31) and (30) it is possible to compare the scaling of the kinetic viscosity to the parameter α.

The Sub-Particle Scale (SPS) model was first introduced by Gotoh *et al.* [38, 39]. The conservation of momentum equation can be written as:

$$\frac{D\mathbf{v}}{Dt} = -\frac{1}{\rho} \nabla P + \mathbf{g} + \nu \nabla^2 \mathbf{v} + \frac{1}{\rho} \nabla \tau \tag{33}$$

and here the laminar term is treated in equation (31) and τ represents the SPS stress tensor. Boussinesq's hypothesis for the eddy viscosity states that the Reynolds stress tensor τ_{ij} is proportional to the trace-less mean strain rate tensor.

$$\frac{\tau_{ij}}{\rho} = \mu_t \left\{ 2S_{ij} - \frac{2}{3} k \delta_{ij} \right\} - \frac{2}{3} C_l \Delta^2 \delta_{ij} |S_{ij}|^2 \tag{34}$$

where τ_{ij} is the sub-particle stress tensor, μ_t is the turbulence eddy viscosity, k the SPS turbulence kinetic energy, C_l is a constant equal to 0.0066 and S_{ij} the element of SPS strain tensor. Here the implementation suggested in [40] is used:

$$\begin{aligned} \frac{Dv_i}{Dt} &= -\sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \cdot \nabla_i W_{ij} \\ &+ \nu \sum_j \frac{4 m_j r_{ij} \cdot \nabla_i W_{ij}}{(\rho_i + \rho_j)(|r_{ij}|^2 + \nu^2)} u_{ij} \\ &+ \sum_j m_j \left(\frac{\tau_i}{\rho_i^2} + \frac{\tau_j}{\rho_j^2} \right) \cdot \nabla_i W_{ij}. \end{aligned} \tag{35}$$

In Monaghan [41] the fluid in the SPH formulation was treated as weakly compressible, and an equation of state was used to determine the pressure in the fluid. The idea behind using this artificial compressibility is to reduce the prohibitively small time steps required to a reasonable level by slowing the speed of sound in the fluid. This reduced speed of sound should, however, be at least an order of magnitude faster than the maximum fluid velocity which keeps the density variations close. Monaghan applied the following equation of state for water to model free surface flows:

$$p = B \left[\left(\frac{\rho}{\rho_0} \right)^\gamma - 1 \right] \tag{36}$$

where γ is a constant taken to be 7 in more circumstances, ρ_0 is the reference density, and B is a problem dependent parameter, which limits the maximum change in density. The subtraction of 1 can remove the boundary effect for free surfaces and it can be seen that a small oscillation in the density may result in a large variation of the pressure. In the current work

$$B = \frac{c_0^2 \rho_0}{\gamma} \tag{37}$$

where c_0 is the speed of sound at the reference density.

The particles are updated using the XSPH technique of Monaghan [42] which was introduced to stop SPH particles pass through each other. The idea is that each particle is moved with an average of the velocities of its neighbours. This reduces or even eliminates the number of particles passing through each other. The method is non-dissipative and conserves linear and angular momentum. This smoothing also has the further advantage of reducing local disorder at

the expense of the energy not being conserved[37].

$$\frac{d\mathbf{r}_i}{dt} = \mathbf{v}_i + \epsilon \sum_j m_j \frac{2}{\rho_i + \rho_j} v_{ji} W_{ij} \tag{38}$$

where ϵ is a user defined parameter usually taken to be 0.5.

Near the boundary and free surfaces, particles have a cut down smoothing kernel due to the absence of neighbouring particles. To correctly handle these conditions the kernel function W_{ij} or its gradient are modified. Two of the possible methods are kernel correction and kernel gradient correction. The kernel correction is investigate by Bonet and Lok [43] and Liu *et al.* [44], while Vaughan [45] investigated the kernel gradient correction. For the kernel correction method, the kernel is changed to enable polynomial functions of a given degree to be interpolated exactly. However, Bonet and Lok considered the linear correction of the kernel to be too computationally expensive, since the parameters needed to ensure that any linear velocity distribution is exactly interpolated depend on $\mathbf{x}$. They suggested to simplify the calculation by using a constant correction

$$v_i = \sum_j \frac{m_j}{\rho_j} v_j W_{ij} / \sum_j \frac{m_j}{\rho_j} W_{ij}. \tag{39}$$

Another option is to modify the kernel gradient used in the equation of motion, as:

$$\widetilde{\nabla} = L_j \nabla W_{ij} \tag{40}$$

$$L_i = M_i^{-1} \tag{41}$$

$$M_i = \sum_j \frac{m_j}{\rho_j} \nabla W_{ij} \otimes (r_i - r_j) \tag{42}$$

It should be noted that when the particle i is away from the boundaries and free surfaces, M_i is equal to the identity matrix and hence no correction is made to the kernel gradient. However, when the particle i is close to a boundary or free surface, the distribution of particles around it does not remain symmetric and the correction is applied. This correction is anisotropic since the off diagonal terms of the L_i involve both spatial coordinates.

In SPH while the simulations are realistic, the pressure field of the particles can exhibit large pressure oscillations. Many approaches have been used to try and reduce the problem. These include correcting the kernel via equation (39) and development of incompressible solvers. However it is also possible to apply a filter over the density of the particles and then use this new smoothed value.

The Shepard filter[46] is a correction which is applied after a user specified number of steps. The correction is as

follows:

$$\rho_i^{new} = \sum_j \rho_j \widetilde{W}_{ij} \frac{m_j}{\rho_j} = \sum_j m_j \widetilde{W}_{ij} \tag{43}$$

where the kernel has been corrected using a zeroth-order correction of equation 39

$$\widetilde{W}_{ij} = W_{ij} / \sum_j \frac{m_j}{\rho_j} W_{ij}. \tag{44}$$

A first order correction called moving least squares (MLS) was first developed by Dilts [47, 48]. Since it is first order, linear variations of the density field can be exactly reproduced. The correction reads:

$$\rho_i^{new} = \sum_j \rho_j W_{ij}^{MLS} \frac{m_j}{\rho_j} = \sum_j m_j W_{ij}^{MLS} \tag{45}$$

where the corrected kernel is

$$W_{ij}^{MLS} = W_{ij}^{MLS}(r_i) = \beta(r_i) \cdot (r_i - r_j) W_{ij}. \tag{46}$$

Time Marching SPH

To perform time-marching simulations each particle is updated using a global fixed time step Δt. For clarity, consider the following system of equation for density, momentum and position:

$$\frac{d\rho_i}{dt} = D_i \tag{47a}$$

$$\frac{d\mathbf{u}_i}{dt} = \mathbf{F}_i \tag{47b}$$

$$\frac{d\mathbf{r}_i}{dt} = \mathbf{V}_i. \tag{47c}$$

If $\mathbf{V}_i$ represents the velocity contribution from particle i only, then $\mathbf{V}_i = \mathbf{v}_i$. However, it can also include the contribution of the neighbouring particles (via the XSPH correction).

The simplest method considered is the semi implicit Euler scheme. The scheme is semi implicit since only the position $\mathbf{r}$ is updated in an implicit manner.

$$\rho_i^{n+1} = \rho_i^n + \Delta t^n D_i^n \tag{48a}$$

$$\mathbf{v}_i^{n+1} = \mathbf{v}^n + \Delta t^n \mathbf{F}_i^n \tag{48b}$$

$$\mathbf{r}_i^{n+1} = \mathbf{r}_i^n + \Delta t^n \mathbf{V}_i^{n+1} \tag{48c}$$

The leap-frog scheme gets it name by updating the positions $\mathbf{r}$ and the velocities $\mathbf{v}$ at interleaved points. The leap-frog scheme is second order in time and is written as:

$$\rho_i^{n+1} = \rho_i^n + \Delta t^n D_i^n \tag{49a}$$

$$\mathbf{v}_i^{n+1/2} = \mathbf{v}^{n-1/2} + \Delta t^n \mathbf{F}_i^n \tag{49b}$$

$$\mathbf{r}_i^{n+1} = \mathbf{r}_i^n + \Delta t^n \mathbf{V}_i^{n+1/2}. \tag{49c}$$

The initial velocity is given by

$$\mathbf{v}_i^{-1/2} = \mathbf{v}_i^0 - \frac{1}{2}\Delta t^0 \mathbf{F}_i^0. \tag{50}$$

The velocity at time step n is required when computing the forces at time step n and can be approximated using the midpoint rule

$$\mathbf{v}_i^n = \frac{1}{2}\left(\mathbf{v}_i^{n-1/2} + \mathbf{v}_i^{n+1/2}\right). \tag{51}$$

The Verlet integration [49] is a very common time integration scheme used in molecular dynamics. The basic idea is to expand two Taylor series for the position $\mathbf{r}_i$, one forward and one backward in time.

$$\mathbf{r}_i^{n+1} = \mathbf{r}_i^n + \mathbf{v}_i^n \Delta t + \frac{1}{2}\mathbf{F}_i^n \Delta t^2 + \frac{1}{6}\mathbf{s}_i^n \Delta t^3 + O(\Delta t^4) \tag{52a}$$

$$\mathbf{r}_i^{n-1} = \mathbf{r}_i^n - \mathbf{v}_i^n \Delta t + \frac{1}{2}\mathbf{F}_i^n \Delta t^2 - \frac{1}{6}\mathbf{s}_i^n \Delta t^3 + O(\Delta t^4) \tag{52b}$$

The scheme employed in this work is split into two parts. Normally the variables are calculated using

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^{n-1} + 2\Delta t^n \mathbf{F}_i^n, \tag{53a}$$

$$\mathbf{r}_i^{n+1} = \mathbf{r}_i^n + \Delta t^n \mathbf{v}_i^n + 0.5(\Delta t^n)^2 \mathbf{F}_i^n, \tag{53b}$$

$$\rho_i^{n+1} = \rho_i^{n-1} + 2\Delta t^n D_i^n. \tag{53c}$$

Since these equations are not coupled, every few iterations (10 to 40) the variables are calculated using the explicit Euler scheme:

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^n + \Delta t^n \mathbf{F}_i^n, \tag{54a}$$

$$\mathbf{r}_i^{n+1} = \mathbf{r}_i^n + \Delta t^n \mathbf{v}_i^n + 0.5(\Delta t^n)^2 \mathbf{F}_i^n, \tag{54b}$$

$$\rho_i^{n+1} = \rho_i^n + \Delta t^n D_i^n. \tag{54c}$$

Symplectic time integration algorithms are designed for the numerical solution of Hamliton's equations and since these

conserve the Hamiltonian and are widely applied in molecular dynamics where long term evolution is required. These schemes are also reversible in the absence of friction or viscous forces[50].

Moving Objects

In the application of SPH, there are two possible types of objects interacting with the fluid. The first have pre-defined movement and the second objects that are moved by the fluid. For the first type, the objects interact with the fluid in such a way that the fluid is displaced by their movement. However, the motion of the object is independent of the fluid that is moving through. Objects of the second type have a two way interaction. For their motion, the equations of rigid body dynamics are required, like for the ditching problem investigated here.
Using Newton's second law the resulting force **F** acting on a rigid body of mass m becomes

$$\mathbf{F} = m\dot{\mathbf{v}}_{cg}. \tag{55}$$

The general moment equation about the centre of gravity is given by

$$G = \dot{\mathbf{h}} \tag{56}$$

where **G** is the resulting moment of the force **F** and **h** is the resulting angular momentum of the body about the centre of gravity. Now, considering the body has angular velocity ω with components ω_x, ω_y, and ω_z

$$\boldsymbol{\omega} = \mathbf{i}\omega_x + \mathbf{j}\omega_y + \mathbf{k}\omega_z \tag{57}$$

the velocity of a mass point of the rotating body becomes

$$\mathbf{V} = \mathbf{V}_{cg} + \boldsymbol{\omega} \times \mathbf{r}. \tag{58}$$

Hence the angular momentum of a rigid body about the centre of gravity is

$$\mathbf{h} = \int \mathbf{r} \times \left(\mathbf{V}_{cg} + \boldsymbol{\omega} \times \mathbf{r}\right) dm = \int \mathbf{r} \times \mathbf{V}_{cg}\ dm + \int \mathbf{r} \times (\boldsymbol{\omega} \times \mathbf{r}) dm = \int \mathbf{r} \times (\boldsymbol{\omega} \times \mathbf{r}) dm. \tag{59}$$

In addition,

$$\int \mathbf{r} \times (\boldsymbol{\omega} \times \mathbf{r}) dm = \int (\boldsymbol{\omega}(\mathbf{r} \cdot \mathbf{r}) - \mathbf{r}(\boldsymbol{\omega} \cdot \mathbf{r}))\ dm = \int (\boldsymbol{\omega}\mathbf{r}^2 - \mathbf{r}(\boldsymbol{\omega} \cdot \mathbf{r})) dm \tag{60}$$

and by substituting $\mathbf{r} = \mathbf{i}x + \mathbf{j}y + \mathbf{k}z$ and equation (60) into equation (59),

$$\mathbf{h} = \boldsymbol{\omega} \int (x^2 + y^2 + z^2) dm - \int \mathbf{r}(x\omega_x + y\omega_y + z\omega_z) dm \tag{61}$$

$$\mathbf{h} = \begin{bmatrix} h_x \\ h_y \\ h_z \end{bmatrix} = \int \begin{bmatrix} (y^2 + z^2) & -xy & -xz \\ -xy & (x^2 + z^2) & -yz \\ -xz & -yz & (x^2 + y^2) \end{bmatrix} dm, = \mathbf{I}\boldsymbol{\omega} \tag{62}$$

where $\mathbf{I}$ is defined as the inertia matrix. The diagonal terms I_{xx}, I_{yy} and I_{zz} are the moments of inertia while the off diagonal terms $-I_{xy}$, $-I_{xz}$ and $-I_{yz}$ are the products of inertia.

When a reference frame is fixed to the body (x_b, y_b, z_b) the inertia matrix remains constant. However the frame of reference now rotates with angular velocity $\boldsymbol{\omega}$. So in the body frame of reference equations (55) and (56) become

$$\mathbf{F} = m\frac{\partial \mathbf{V}_{cg}}{\partial t} + m\boldsymbol{\omega} \times \mathbf{V}_{cg} \tag{63}$$

and

$$\mathbf{G} = \frac{\partial \mathbf{h}}{\partial t} + \boldsymbol{\omega} \times \mathbf{h}. \tag{64}$$

If the forces and moments are transformed into the body reference frame (x_b, y_b, z_b) $\mathbf{F} = \mathbf{i}F_{x_b} + \mathbf{j}F_{y_b} + \mathbf{k}F_{z_b}$ and $\mathbf{G} = \mathbf{i}G_{x_b} + \mathbf{j}G_{y_b} + \mathbf{k}G_{z_b}$ this yields

$$\begin{aligned} F_{x_b} &= m\left(\dot{u} + \omega_y w - \omega_z v\right) \\ F_{y_b} &= m\left(\dot{v} + \omega_z u - \omega_x w\right) \\ F_{z_b} &= m\left(\dot{w} + \omega_x v - \omega_y u\right) \end{aligned} \tag{65}$$

and

$$\begin{aligned} G_{x_b} &= \dot{h}_x + \omega_y h_z - \omega_z h_y \\ G_{y_b} &= \dot{h}_y + \omega_z h_x - \omega_x h_z \\ G_{z_b} &= \dot{h}_z + \omega_x h_y - \omega_y h_x \end{aligned} \tag{66}$$

$$
\begin{aligned}
G_{x_b} &= I_{xx}\dot{\omega}_x + (I_{zz} - I_{yy})\omega_y\omega_z - I_{xy}(\dot{\omega}_y - \omega_x\omega_z) \\
&- I_{xz}(\dot{\omega}_z - \omega_x\omega_y) - I_{yz}(\omega_y^2 - \omega_z^2) \\
G_{y_b} &= I_{yy}\dot{\omega}_y + (I_{xx} - I_{zz})\omega_x\omega_z - I_{xy}(\dot{\omega}_x - \omega_y\omega_z) \\
&- I_{xz}(\omega_x^2 - \omega_z^2) - I_{yz}(\dot{\omega}_z - \omega_x\omega_y) \\
G_{z_b} &= I_{zz}\dot{\omega}_z + (I_{yy} - I_{xx})\omega_x\omega_y - I_{xy}(\omega_x^2 - \omega_y^2) \\
&- I_{xz}(\dot{\omega}_x - \omega_y\omega_z) - I_{yz}(\dot{\omega}_y - \omega_x\omega_z)
\end{aligned}
\tag{67}
$$

Particle Interactions

In general, the support of the kernel function is compact and only a finite number of particles are within this support. Three possible ways to calculate the nearest neighbours are discussed below: the all-pair search, the linked-list search algorithm and the tree-search algorithm. The all-pair search, or brute force method, is a direct and simple algorithm. For any given particle i, the distance r_{ij} to each particle j is calculated. If the distance r_{ij} is smaller than the dimension of the support for i then the particles i and j interact. This search is carried out on all N particles and so $O(N^2)$ operations are required. Hence the all-pair search is only computationally efficient if the total number of particles is very small.

The linked-list search algorithm works best for cases where the support radius is constant across all particles. It was shown by Monaghan and Gingold [51] that by using cells as a book-keeping device the computational cost of particles interactions could be reduced. If all particles are assigned to bins and identified through a linked-list, the computational time is reduced as only certain bins need to be checked. A temporary mesh is overlaid on the problem domain. The mesh spacing is selected to match the dimension of the support domain. Then for the particle i, its nearest neighbouring particles can only be in the same grid cell or in adjoining cells. Domínguez *et al.* [52] compared the performance for the two different methods to create list of neighbours namely the Cell-Linked List (CLL) and the Verlet List (VL). The CLL method sets up a list linked to every cell and is the method shown in figure 1. The VL method creates a linked list for each vertex. This is usually implemented by generating a simple linked list to contain which particles are in each cell, and a one dimensional array describing which cell each particle is in. They also looked into renumbering the particles so they are "close" in memory for better cache usage. They concluded that for parallel computations VL was better.

The main drawback of the linked-list search algorithm is when a variable smoothing length h is used. In this case the mesh spacing used to define the bin may not be optimal for every particle and hence the efficiency drops. This problem is overcome by using a tree search algorithm. Order trees are created according to particle position which are then searched to find the nearest neighbour particles. The tree method recursively splits the domain until only a single

particle is in each leaf. The search is performed by centering a cube on the particle and checking the overlap of this cube with the volume represented by the node. Finally a check is required to see if the particle is in the support domain. The complexity of a tree search algorithm is order $N \log N$. The ditching calculations use a fixed smoothing length and so the Cell-Linked List is was used in this work.

SPH Formulation

An open source version of an SPH solver is DualSPHysics [27, 28, 53] that was used in this work. A simple flow-chart of the employed SPH method can be seen in figure 3. The formulation for floating objects within DualSPHysics is based on the work of Monaghan *et al.* [54] and examples in the literature have shown a good agreement between the SPH results and experimental data. Further, the standard SPH method suffers from a lack of stability and hence uses an artificial viscosity term π_{ij} or by applying a density renormalisation. Both these fixes help to increase the regularity of the pressure field within the computational domain. However, as can been seen from figure 4 even with options used there are still defects in the pressure field making the pressure calculation at a point a non trivial task. Colagrossi [55] improved this pressure field via using a second order accurate interpolation with moving least square kernel.

Results and Discussion

Simple Cases

Simple flow cases were initially considered with SPH to allow tuning of the various method parameters and assess the effect of different boundary conditions on the obtained results. The SPH method, requires careful use and systematic assessment of their numerical parameters since otherwise the obtained results may violate the conservation laws and lead to solutions with incorrect physics. Only some of the studies conducted in the preparation of the SPH method for helicopter ditching are presented in this work.

Effects of applying smoothing to an idealised problem

The effect of different filters to improve the smoothness of the solution was first investigated. Figures 5 and 6 show the effect of the Shepard[46] and the moving Least-Squares[47, 48] filters on the obtained particle density for the case of a cube dropped on the surface of water. The results suggest that for the cases of ditching filtering of the solution may be necessary to smooth out the pressure oscillations but the global behaviour of the object remains largely unchanged. The Shepard filter applied every 20 iterations was chosen since it smooths out the density and pressure oscillations with the least amount of computational overhead.

Effect of boundary condition on the solution to an idealised problem

In addition to the solution smoothness, higher frequency oscillations may be present on the force and accelerations of floating bodies as a result of the applied boundary condition between the fluid particles and the particles attached to floating objects. The dynamic boundary condition of Dalrymple [56] was first assessed. According to this condition, boundary particles are forced to satisfy the same equations as fluid particles. However, they do not move freely and so remain fixed in position, unless their position changes due to some external function, or rigidly under loading for floating objects. The repulsive boundary condition can also be used [57]. This condition uses a repulsion function to ensure that a fluid particle can never cross a solid boundary. This is a much more involved boundary condition which also requires the geometric normals for every point on the boundary.

Figure 7 shows the results of a two dimensional cube drop. As can be seen, the Dalrymple boundary condition produces a large oscillation in the velocity of the cube where the repulsive force boundary condition does not. This also leads to a much higher deceleration in the initial phase of the impact with smaller oscillations at later times. Therefore, the repulsive boundary condition looks more attractive, provided that the complex task of computing the surface normals for each particle can be performed. This task is trivial for a simple object with flat surfaces but can be harder for the case of a ditching helicopter due to the complex fuselage surface, and its representation as a cloud of points with little or no connectivity information.

Choice of lattice type for floating objects and the fluid

Different lattice types can be used to represent the boundary and the fluid domain and the surface of a floating object. This leads to a total of four possible combinations. The two different lattice types are shown in figure 8. The type-1 lattice has just a single row of particles representing the object while the type-2 lattice has a double row. For a given weight of object the type-2 lattice particles will have half the mass of the type-1. The type-2 lattice will also roll over quicker since, in effect, the cube had two of its corners rounded off and hence the forces on either side of the square will lead to a moment causing the object to roll. The reason why a double layer of lattice is normally used for objects is that it is much harder for the fluid particles to penetrate the boundary walls.

Due to the importance of the drop height on the observed pressure fluctuations, a simple test case was used of dropping a $10 \times 10 \times 10$m cube into the middle of a 30m square tank containing water 15 meters deep from a height of 1 meter. The density of the cube is half that of the water and hence its equilibrium position will have half of the cube sitting out of the water. The final height should be at 15.55 m due to the small size of the tank making the displaced water increase the water level by about four percent. Figure 9 shows the four different combinations of lattice. The two cases where the fluid and boundary have the same type of lattice give very consistent results but the final position of the cube is about 1.5 m too high. The normal type-2 lattice boundary, and type-1 lattice fluid in this case, gave a better final position at around 15m. However the type-1 lattice boundary with a type-2 lattice fluid did something

very different. For the first four seconds it run correctly even if it resulted with at a much higher position in the water. However, after this time fluid particles start leaking from the domain causing the height to drop. This is more clearly seen in the velocities as this configuration is not converging to zero.

Figure 10 shows the effect of increasing the number of particles. For the type-1 lattice boundary and fluid the solution becomes more oscillatory. However, the final equilibrium position was only 0.4m too high. The normal type-2 lattice boundary and type-1 lattice fluid nearly hit the bottom of the tank in this case, and also showed a drop in the centre of gravity equilibrium under particles refinement, drifting further away from the correct answer. Extrapolating the results it would appear that the type-2 lattice boundary and type-1 lattice fluid will have an equilibrium position approximately as if the cube was of the same density as that of the water. The type-1 lattice boundary and fluid gave the correct behaviour under particle refinement and the particles did not leak. It was therefore used for helicopter ditching simulations.

Obtaining the correct equilibrium position with coarse particle density

As discussed in the previous section, even when the masses of the fluid and the body were correct the body did not have the correct buoyancy. This was because the body displaced too much fluid and hence ended floating too high in the water. Consider a tank of 3m ×3m×3m with 2m of fluid in it. The results can be seen in table 1 for different particles sizes. As the particle size is reduced the mass of the fluid in the container converged to the modelled condition (18,000kg). This is because particles are not placed exactly on the tank walls. This means that in the 0.1 particle case, instead of $30 \times 30 \times 20$ particles the SPH method is started with $29 \times 29 \times 19 = 15979$. For a particle size of 10cm there is a 11% error. This error scales linearly with the particles size so, at a 1cm scale the error reduced to one percent.

Something similar happens when a fully submerged floating body is added to the tank. Particles are now placed on the faces of the cube. So for lattice type-1, the number of particles in the floating object is:

$$6 \times (n-1)^2 + 12 \times (n-1) + 8, \tag{68}$$

where n is the size of the cube divided by the particle size. The number of displaced fluid particles is:

$$(n+1)^3. \tag{69}$$

For the buoyancy to be correct the amount of displaced fluid has to equal 1000kg which is n^3 particles and hence the error is

$$\frac{3n^2 + 3n + 1}{n^3} \tag{70}$$

which again is order n^{-1}. The results of this can be seen in table 2. For 10cm particles size a one meter cubed object

will displace 1331kg of fluid and hence buoyancy would cause the cube to move upwards. In this case the equilibrium position would have about 25cm of cube height above the water line. As the particle size is reduced this discrepancy is reduced linearly. The difference is slightly smaller for partially submerged objects as shown in table 3 because the discrepancy due to the upper surface has been removed.

Vertical half-buoyant cylinder drop

SPH method validated against the experiments of Greenhow and Lin[58] for the entry of a half-buoyant solid cylinder into calm water. The experimental setup is shown in figure 11(a), a cylinder of density of $500kg/m^3$ was allowed to fall freely from the height of $0.8m$ under gravity acceleration; the water depth was $0.3m$. The viscosity between the cylinder SPH particles and fluid particles was neglected. Five cases were compared with different distances d between the particles. The penetration depth of the cylinder for all cases and experimental results are shown in figure 11(b). A more detailed discussion of this validation case can be found in [59].

Using complex geometries

Using simple geometric shapes like a cube is easy with either lattice type onto the body. However, for complex general cases, this is a non trivial problem. Consider for example an approximate fuselage of the full scale AW101 helicopter shown in figure 12. Four different particle resolutions, 20cm, 10cm, 5cm and 2cm, were used to represent the fuselage and the results can be seen in figures 13. Firstly a grid of particles is set up with the correct spacing. If the point is greater than half the particle resolution away from the surface then the point is discarded, otherwise it is kept. This means that for any give surface the particles may over- or under-approximate it by half their resolution. This effect can be seen in the closeup view near the radar dome and the bottom of the fuselage (bottom of figure 13). For the 20cm resolution (black squares) the points lie outside the radar dome and inside the bottom of the fuselage making the effective height of the radar dome bigger whereas in the $10cm$ resolution case this is reversed. The other drawback of this method is that surfaces with curvature will be represented by straight line segments. Even the 2cm resolution in the high curvature region of the radar dome has a very pronounced "staircase" effect. At present, the surface particle generation method can work with STL files produced by standard CAD systems, and can extract a representation of any helicopter fuselage suitable for ditching computations with the SPH method.

Demonstration of SPH for Helicopter Ditching

The representation of the AW159 fuselage in the format used for ditching computations can be seen in figure 14. Based on the discussion of the previous paragraph a resolution of 5cm for the full size fuselage generates a surface with $77,000$ particles. The figure shows the complexity of the problem at hand, and the flexibility of the method.
Validation of the SPH method was then carried out against experiments, conducted at both the basins of DGA/TH (Val

de Reuil) and ECN (Nantes) for a 1/10th scaled model of the AW159 fuselage[24]. The experiments provided data for the motion of the model as well as pressure and accelerometer readings from a few points on the model. Figure 15 shows the employed 1/10th scale model as well as a still photograph of the model 0.2 seconds after a drop on the surface of water at sea-state zero, and with the main rotor providing 67% of lift. This particular condition has also been simulated using the SPH method.

To account for the vertical lift in the experiment a forcing was applied on the rotor hub to match that of the experiment. The forcing was continued after the initial impact to mimic the experiment. The main effect of this forcing is to reduce the effect of gravity on the fuselage and it the reason why the initial part of the drop does not accelerate at g.

Table 4 shows the computational cost for four different particle spacing. The coarsest spacing 20cm full scale (2cm model scale) contains 76 points along the fuselage while the finest has 509. As the particle spacing is halved the computational cost increases by a factor of 16 since it takes twice as many time-steps to reach the same time in simulation for a fixed CFL number. The 0.5cm model scale spacing results in a runtime of 21 hours for each second of simulated time, on a 4-core Intel Xeon CPU running at 3.30GHz.

Figure 16 shows the pressure on the fuselage at different times during ditching. The pressure is scaled with the maximum value seen during the run. It can be seen that the water line does not reach very far up the fuselage and hence most of the fuselage has zero pressure on it. However, as can be seen from Figure 17 the comparison with the vertical velocity between the experimental data and the SPH simulation is good. The results show that the velocity and acceleration are predicted fairly well if the correct size of particles is used. The SPH results also appears to capture well the peaks of the vertical acceleration and velocity with some noise present in the solution. The acceleration in particular, is reasonably well predicted apart from the initial impact for the 67% rotor lift case.

Figure 18 shows the AW159 fuselage drop at sea state 4 where the regular waves are 4 meters high with a wave slope of 0.1. The particle size was 1.5cm on the model scale which is larger than ideal but still required 24 hours of CPU for 2.6 seconds of real-time simulation. The figure shows that the fuselage sits ever so slightly high in the water. This is due to the fact that the rotor model is active during the whole of the computation producing lift in the vertical direction meaning, in effect, that the fuselage only has about 1/3 of its weight. Another reason for the high position is the larger particle size.

Figure 19 shows the motion of the fuselage more clearly. There is almost a 2G impact in the Z direction and one quarter of this in the X direction. The initial vertical velocity looks very similar to a ditch into sea state zero but the effects of the waves can be clearly seen. The size of the waves is slightly too big, and the fuselage moves slowly towards the beach over time. The roll rate is also increasing but this is due to the fact that the fuselage sits relatively high in the water.

Conclusions and Future Work

In this paper SPH has been demonstrated for helicopter ditching. The method is mesh-free and in comparison to traditional CFD methods appears to be easier to use due to the lack of the mesh-generation step. On the other hand, the results of SPH depend heavily on the use of appropriate particle resolution, flow model parameters, and correct boundary conditions between the solid and fluid particles. Simple cases like the drop of a cube on the surface of water were initially used for the investigation of the effects of all the aforementioned parameters.

Once the effect of the boundary conditions and flow model parameters were quantified, the simulation of an AW159 ditching was attempted. The case of a vertical drop of the fuselage at sea-state zero was attempted and the results showed good agreement with the measured mean values regarding the velocity, acceleration and position of the fuselage versus time. Further cases included a vertical drop on the crest of a wave to demonstrate the potential of the method.

Overall, SPH was found satisfactory for the ditching task even though some user experience and careful selection of the model parameters were necessary. In the future, efforts will be directed towards establishing a practical list of criteria for the selection of the numerical parameters of the SPH method so that routine analyses of helicopter ditching can be performed.

Acknowledgements

The financial support via the NGVL project of AgustaWestland and the Business Innovation and Skills Department of UK is gratefully acknowledged.

References

[1] Authority, C. A., "Summary Report on Helicopter Ditching and Crashworthiness Research," Tech. Rep. CAA Paper 2005/06, Civil Aviation Authority, 2005.

[2] Chen, C. C., Muller, M., and Fogarty, M., K, "Rotorcraft ditchings and water-related impacts that occurred from 1982 to 1989 – Phase I," Tech. Rep. DOT/FAA/CT-92/13, U.S. Department of Transportation, Federal Aviation Administration, 1993.

[3] Muller, M., and Bark, L. W., "Rotorcraft ditchings and water-related impacts that occurred from 1982 to 1989 – Phase II," Tech. Rep. DOT/FAA/CT-92/14, U.S. Department of Transportation, Federal Aviation Administration, 1993.

[4] "CAST Crashworthiness of Helicopters on Water: Design of Structures Using Advanced Simulation Tools," Project G4RD-CT-2000-00178 partially funded by the European Union under the Aeronautics part of the FP5-GROWTH RTD, 2000-2003. `https://cordis.europa.eu/project/rcn/52827_en.html` (accessed October 2018).

[5] Pentecôte, N., and Vigliotti, A., "Crashworthiness of helicopters on water: Test and simulation of a full-scale WG30 impacting on water," *International journal of crashworthiness*, Vol. 8, No. 6, 2003, pp. 559–572.

[6] Delsart, D., "GARTEUR HC/AG-15: Improvement of SPH methods for application to helicopter ditching," , 2007-

2010. http://www.garteur.org/Action%20Group%20posters/HC_AG-15_Poster_v1_feb11.pdf (accessed October 2018).

[7] Francesconi, E., and Anghileri, M., "Towards a Methodology to Design Water Impact Crashworthy Structures," 66^{th} Annual Forum of the American Helicopter Society, Phoenix, Arizona, USA, 11-13 May 2010. Paper No. 82.

[8] Toso-Pentecôte, N., Delsart, D., Vagnot, A., and Kindervater, C., "Evaluation of Smooth Particle Hydrodynamic methods for the simulation of helicopter ditching," 66^{th} Annual Forum of the American Helicopter Society, Phoenix, Arizona, USA, 11-13 May 2010. Paper No. 138.

[9] "Smart Aircraft in Emergency Situations," Project 266172 partially funded by the European Union under FP7-TRANSPORT, 2011-2014. https://cordis.europa.eu/project/rcn/97150_en.html (accessed October 2018).

[10] Korobkin, A. A., "Wagner-type models of water impact with separation for a finite wedge," *Proceedings of the 28th International workshop on Water Waves and Floating Bodies*, 2013, pp. 117 – 120.

[11] Tassin, A., Korobkin, A., and Cooker, M., "On analytical models of vertical water entry of a symmetric body with separation and cavity initiation," *Applied Ocean Research*, Vol. 48, 2014, pp. 33 – 41.

[12] Siemann, M. H., Schwinn, D. B., Scherer, J., and Kohlgrüber, D., "Advances in numerical ditching simulation of flexible aircraft models," *International Journal of Crashworthiness*, Vol. 23, No. 2, 2018, pp. 236–251.

[13] von Karman, T., "The impact of seaplanes floats during landing," Tech. Rep. NASA Technical note no 321, NASA, 1929.

[14] Wagner, H., "Phenomena Associated with Impacts and Sliding on Liquid Surfaces," *Journal of Applied Mathematics and Mechanics*, Vol. 12, 1932, pp. 193–235.

[15] Wagner, H., "Landing of seaplanes," Tech. Rep. NASA Technical Memorandum no 622, NASA, 1932.

[16] Chuang, S. L., "Theoretical investigations on slamming of cone-shaped bodies," *Journal of Ship Research*, Vol. 13, 1969, pp. 276–283.

[17] Scolan, Y., and Korobkin, A. A., "Three-dimensional theory of water impact. Part 1. Inverse Wagner problem," *Journal of Fluid Mechanics*, Vol. 440, 2001, pp. 293–326.

[18] Scolan, Y., and Korobkin, A. A., "Energy distribution from vertical impact of a threedimensional solid body onto the flat free surface of an ideal fluid," *Journal of Fluids and Structures*, Vol. 17, 2003, pp. 275–286.

[19] Korobkin, A. A., "Analytical models of water impact," *European Journal of Applied Mathematics*, Vol. 15, 2004, pp. 821–838.

[20] Logvinovichi, G. V., *Hydrodynamics of Flows With Free Boundaries*, Naukova Dumka Publishing House, Kiev, Ukraine, 1969.

[21] Korobkin, A., and Malenica, S., "Modified Logvinovich model for hydrodynamic loads on asymmetric contours entering water." *Proc. 20th Intern. Workshop on Water Waves and Floating Bodies*, 2005.

[22] Zhao, R., and Faltinsen, O. M., "Water entry of two-dimensional bodies," *Journal of Fluid Mechanics*, Vol. 246, 1993, pp. 593–612.

[23] Tassin, A., Jacques, N., El Malki Alaoui, A., Nême, A., and Leblé, B., "Assessment and comparison of several analytical models of water impact," *International Journal of Multiphysics*, Vol. 4, No. 2, 2010, pp. 125–140.

[24] Scrase, N., "AW159 Fuselage Ditching Campaign," private communication, 2013.

[25] Lucy, L. B., "A Numerical Approach to Testing the Fission Hypothesis," *The Astronomical Journal*, Vol. 82, No. 12, 1977, pp. 1013–1924.

[26] Gingold, R. A., and Monaghan, J. J., "Smoothed Particle hydrodynamic: theory and application to non-spherical stars," *Monthly Notices of the Royal Astronomical Society*, Vol. 181, 1977, pp. 375–389.

[27] Gomez-Gesteira, M., Rogers, B., Crespo, A., Dalrymple, R., Narayanaswamy, M., and Dominguez, J., "SPHysics - development of a free-surface fluid solver - Part 1: Theory and formulations," *Computers & Geosciences*, Vol. 48, 2012, pp. 289 – 299.

[28] Gomez-Gesteira, M., Crespo, A., Rogers, B., Dalrymple, R., Dominguez, J., and Barreiro, A., "SPHysics - development of a free-surface fluid solver - Part 2: Efficiency and test cases," *Computers & Geosciences*, Vol. 48, 2012, pp. 300 – 307.

[29] Badcock, K., Richards, B., and Woodgate, M., "Elements of computational fluid dynamics on block structured grids using implicit solvers," *Progress in Aerospace Sciences*, Vol. 36, No. 5, 2000, pp. 351–392.

[30] Steijl, R., Barakos, G., and Badcock, K., "A framework for CFD analysis of helicopter rotors in hover and forward flight," *International Journal for Numerical Methods in Fluids*, Vol. 51, No. 8, 2006, pp. 819–847.

[31] Johnson, G. R., Stryk, R. A., and Beissel, S. R., "SPH for high velocity impact computations," *Computer Methods in applied mechanics and engineering*, Vol. 139, 1996, pp. 347–373.

[32] Monaghan, J. J., and Lattanzio, J. C., "A Refined Method for Astrophysical Problems," *Astronomy and Astrophysics*, Vol. 149, 1985, pp. 135–143.

[33] Wendland, H., "Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree," *Advances in Computational Mathematics*, Vol. 4, No. 1, 1995, pp. 389–396.

[34] Monaghan, J. J., "Why particle methods work," *SIAM Journal of Scientific and Statistical Computing*, Vol. 3, No. 4, 1982, pp. 422–433.

[35] Monaghan, J. J., "Particle Methods for Hydrodynamics," *Computer Physics Reports*, Vol. 3, 1985, pp. 71–124.

[36] Monaghan, J. J., "An introduction to SPH," *Computer Physics Communications*, Vol. 48, 1988, pp. 89–96.

[37] Monaghan, J. J., "Smoothed Particle Hydrodynamics," *Annual review of Astronomy and Astrophysics*, Vol. 30, 1992, pp. 543–574.

[38] Gotoh, H., Shibahara, T., and Sakai, T., "Sub-particle-scale turbulence model for the MPS method - Lagrangian flow model for hydraulic engineering," *Advanced Methods for Computational Fluid Dynamics*, Vol. 9-4, 2001, pp. 339–347.

[39] Gotoh, H., Shao, S., and Memita, T., "SPH-LES model for numerical investigation of wave interactions with partially immersed breakwater," *Coastal Engineering Journal*, Vol. 46, 2004, pp. 39–63.

[40] Dalrymple, R. A., and Rogers, B. D., "Numerical modeling of water waves with the SPH method," *Coastal Engineering*, Vol. 53, No. 2-3, 2006, pp. 141–147.

[41] Monaghan, J. J., "Simulating free surface flows with SPH," *Journal of Computational Physics*, Vol. 110, 1994, pp. 399–406.

[42] Monaghan, J. J., "On the problem of penetration in particle methods," *Journal of Computational Physics*, Vol. 82, No. 1, 1989, pp. 1–15.

[43] Bonet, J., and Lok, T. S. L., "Variational and momentum preservation aspects of Smooth Particle Hydrodynamic formulations," *Computer Methods in applied mechanics and engineering*, Vol. 180, No. 1-2, 1999, pp. 97–115.

[44] Liu, W. K., Li, S. F., and Belytschko, T., "Moving least squares Galerkin methods (I) methodology and convergence," *Computer Methods in Applied Mechanics and Engineering*, Vol. 143, 1997, pp. 113–154.

[45] Vaughan, G. L., Healy, T. R., Bryan, K. R., Sneyd, A. D., and Gorman, R. M., "Completeness, conservation and error in SPH for fluids," *International Journal for Numerical Methods in Fluids*, Vol. 56, No. 1, 2008, pp. 37–62.

[46] Shepard, D., "A Two-dimensional Interpolation Function for Irregularly-spaced Data," *Proceedings of the 1968 23rd ACM National Conference*, ACM, New York, NY, USA, 1968, pp. 517–524.

[47] Dilts, G. A., "Moving least-squares particle hydrodynamics I: consistency and stability," *International Journal for Numerical Methods in Engineering*, Vol. 44, No. 8, 1999, pp. 1115–1155.

[48] Dilts, G. A., "Moving least-squares particle hydrodynamics II: conservation and boundaries," *International Journal for Numerical Methods in Engineering*, Vol. 48, No. 10, 2000, pp. 1503–1524.

[49] Verlet, L., "Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules," *Physical Review*, Vol. 159, 1967, pp. 98–103.

[50] Leimkuhler, B. J., Reich, S., and Skeel, R. D., "Integration methods for molecular dynamics," *IMA Voulmes in Mathematics and its applications*, Springer Verlag, 1996, pp. 161–186.

[51] Monaghan, J. J., and Gingold, R. A., "Shock simulation by the particle method SPH," *Journal of Computational Physics*, Vol. 52, No. 2, 1983, pp. 374–389.

[52] Domínguez, J. M., Crespo, A. J. C., Gómez-Gesteira, M., and Marongiu, J. C., "Neighbour lists in smoothed particle hydrodynamics," *International Journal for Numerical Methods in Fluids*, 2010.

[53] Crespo, A. C., Dominguez, J. M., Barreiro, A., Gómez-Gesteira, M., and Rogers, B. D., "GPUs, a New Tool of Acceleration in CFD: Efficiency and Reliability on Smoothed Particle Hydrodynamics Methods," *PLoS ONE*, Vol. 6, No. 6, 2011.

[54] Monaghan, J. J., Kos, A., and Issa, N., "Fluid motion generated by impact," *Journal of waterway, port, coastal and ocean engineering*, Vol. 129, 2003, pp. 250–259.

[55] Colagrossi, A., and Landrini, M., "Numerical simulation of interfacial flows by Smoothed Particle Hydrodynamics," *Journal of Computational Physics*, Vol. 191, 2003, pp. 448–475.

[56] Crespo, A. J. C., Gómez-Gesteira, M., and Dalrymple, R. A., "Boundary Conditions Generated by Dynamic Particles in SPH Methods," *Computers, Materials, & Continua*, Vol. 5, No. 3, 2007, pp. 173–184.

[57] Rogers, B. D., and Dalrymple, R. A., "SPH modeling of tsunami waves," *Advanced Numerical Models for Simulating Tsunami Waves and Runup*, 2008, pp. 75–100.

[58] Greenhow, M., and Lin, W.-M., "Nonlinear-free Surface Effects: Experiments and Theory," Tech. Rep. Technical Report 83-19, MASSACHUSETTS INST OF TECH CAMBRIDGE DEPT OF OCEAN ENGINEERING, 1983.

[59] Leble, V., "Integrated simulation of off-shore wind turbine," Ph.D. thesis, Aerospace Sciences, School of Engineering, University of Glasgow, 2016.

Table 1 Number of particles in 18 cubic meters of water.

Particle size(m)	Number of fluid particles	Mass of particle	Total Mass of fluid (kg)	Percentage error
0.1	15979	1.000	15979	11.2
0.05	135759	0.125	16970	5.7
0.02	2197899	0.008	17583	2.3
0.01	17790799	0.001	17791	1.16

Table 2 Mass of displaced fluid for a fully submerged $1 \times 1 \times 1$m object.

Particle size (m)	Number of fluid particles in tank	Mass of particle	Number of body particles	Particles displaced (kg)	Mass displaced	Percentage error
0.1	15979	1.000	602	1331	1331	33.1
0.05	135759	0.125	2402	9261	1158	15.8
0.02	2197899	0.008	15002	132651	1061	6.1
0.01	17790799	0.001	60002	1030301	1030	3.0

Table 3 Mass of displaced fluid for a partially (20%) submerged $1 \times 1 \times 1$m object.

Particle size (m)	Number of fluid particles in tank	Mass of particle	Number of body particles	Particles displaced	Mass displaced (kg)	Percentage error
0.1	15979	1.000	602	242	242	21.0
0.05	135759	0.125	2402	1764	221	10.5
0.02	2197899	0.008	15002	26010	208	4.0
0.01	17790799	0.001	60002	204020	204	2.0

Table 4 AW159 Ditching drop with different model scale particles spacing (full scale in brackets) on 4 cores of a Xeon E3-1245 at 3.30GHz

Particle spacing	Number of particles			Memory MegaBytes	CPU time per real time (s)
	Total	Boundary	Fuselage		
2cm (20cm)	83k	19k	5k	22	330
1cm (10cm)	576k	76k	19k	153	4,500
0.5cm (5cm)	4319k	303k	77k	1160	72,300
0.3cm (3cm)	19540k	842k	215k	5005	586,000

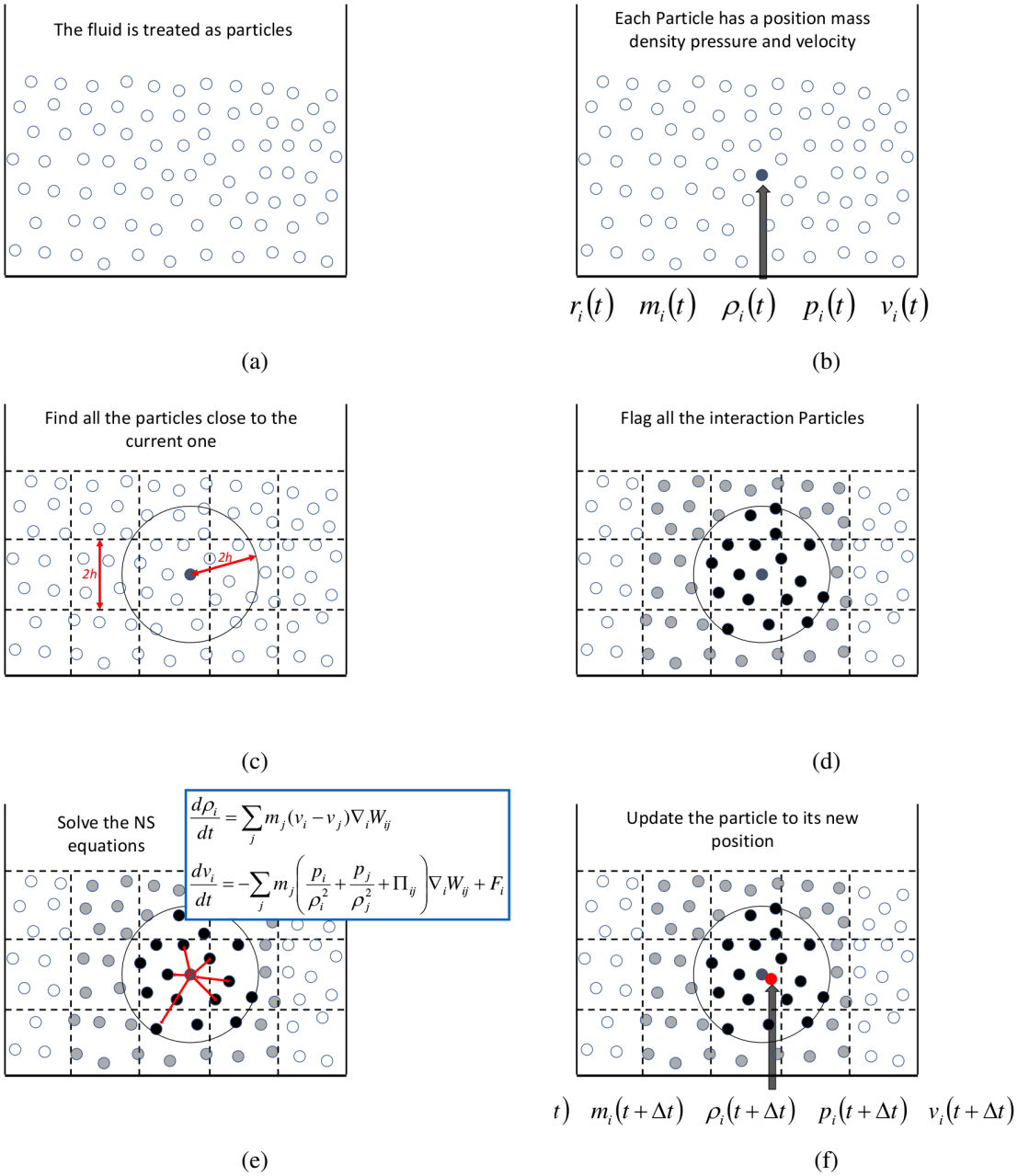


Fig. 1 Overview of the SPH method.

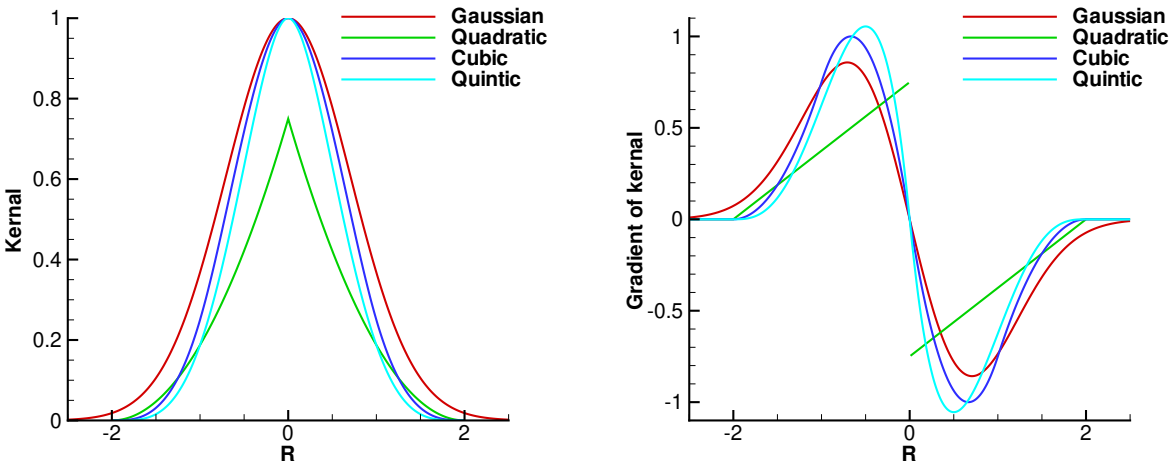


Fig. 2 Example of four commonly used kernels and their gradients used in SPH methods.

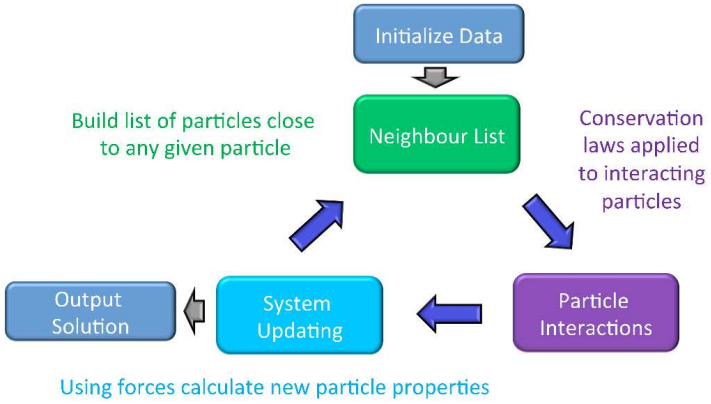


Fig. 3 Schematic of the SPH code.

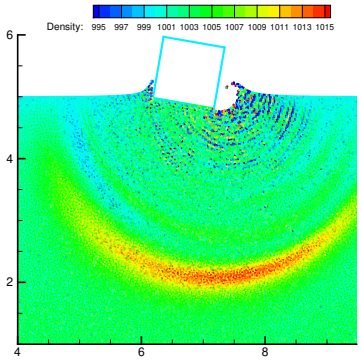
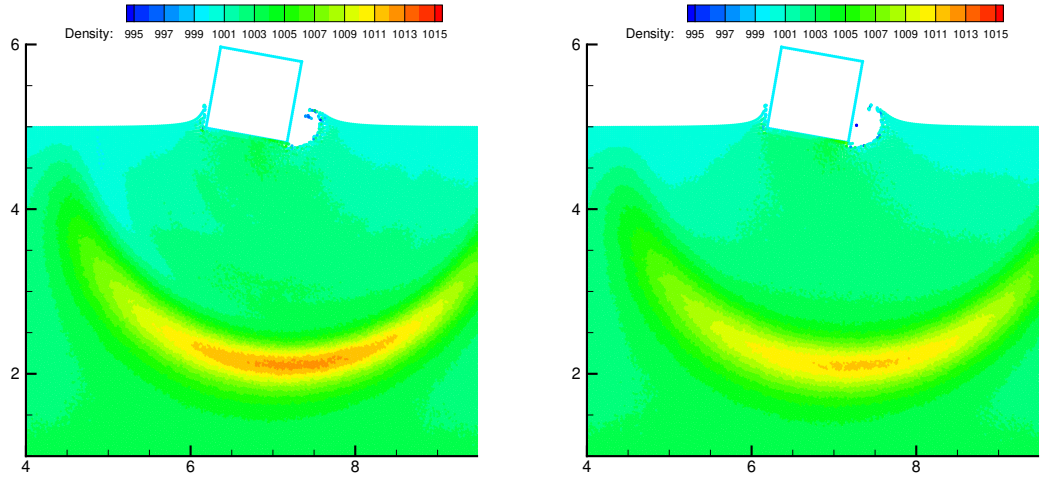
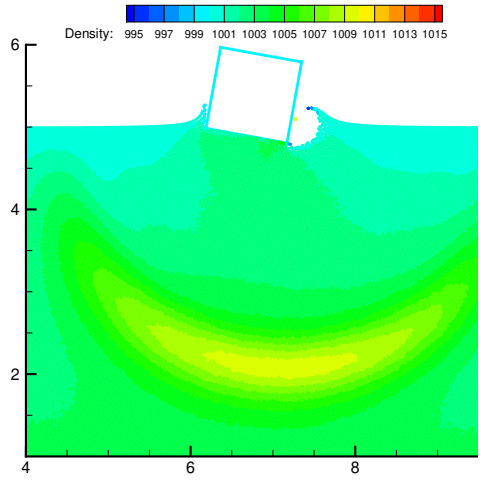


Fig. 4 The non-regular density field in an SPH computation for the case of a cube dropped on the surface of water. Density contours (kg/m^3) are presented

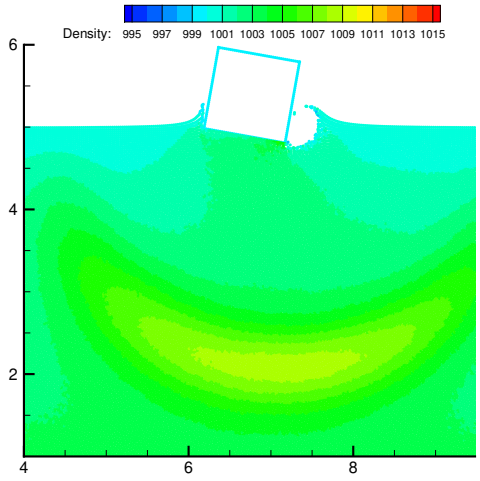


40 steps (a)

20 steps(b)

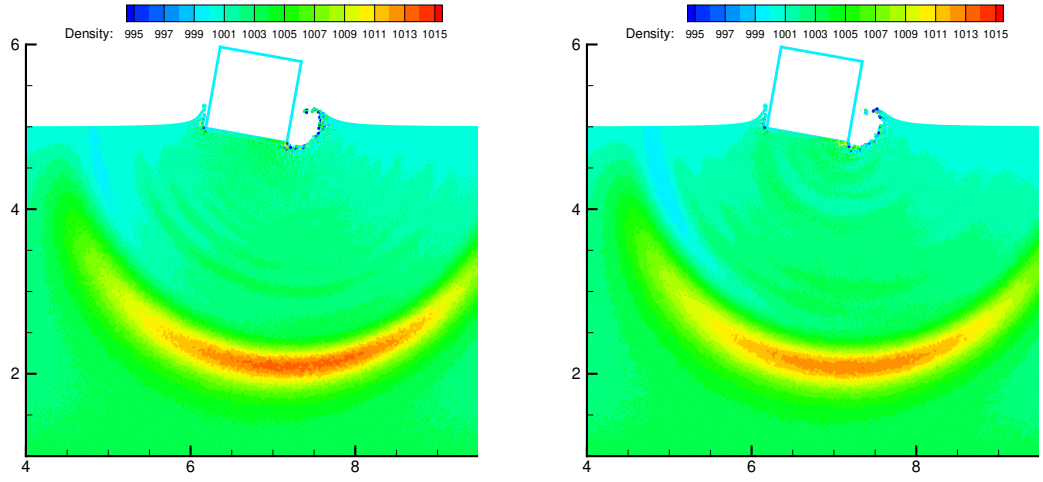


10 steps (c)



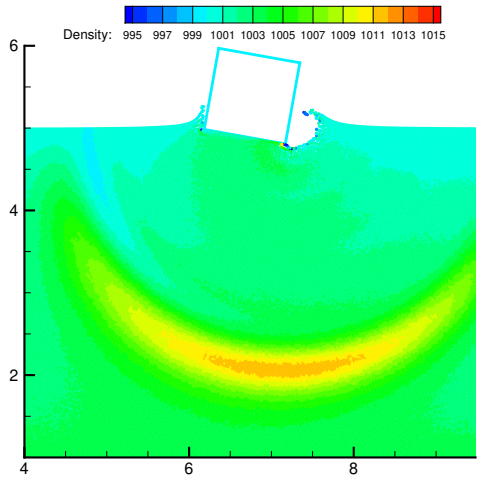
5 steps (d)

Fig. 5 Effect of the Shepard filter on the smoothness of the solution. Density contours (kg/m^3) are presented for a cube dropped on the surface of water.



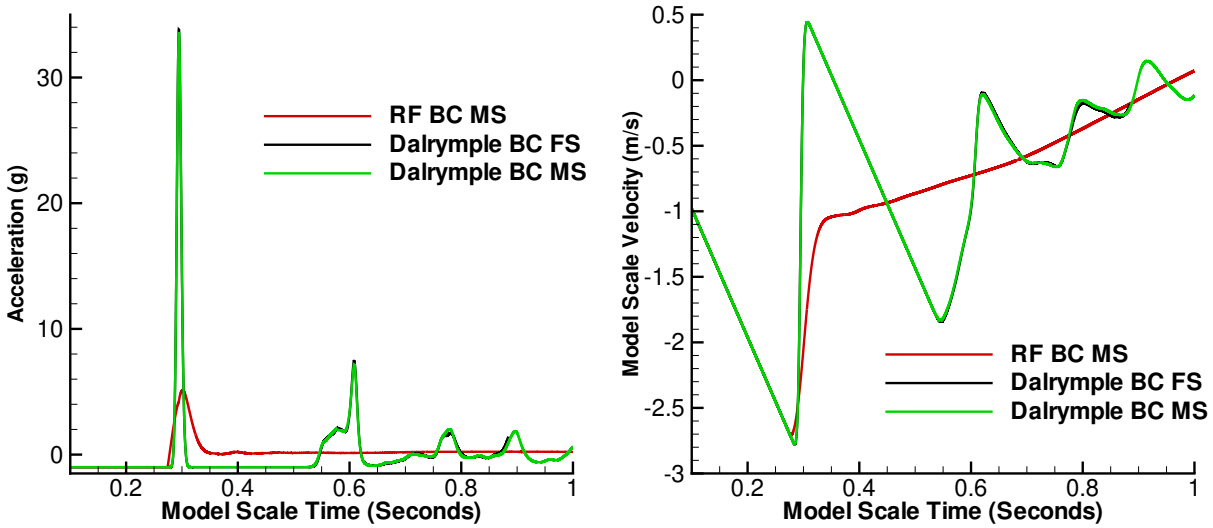
(a) 20 steps

(b) 10 steps



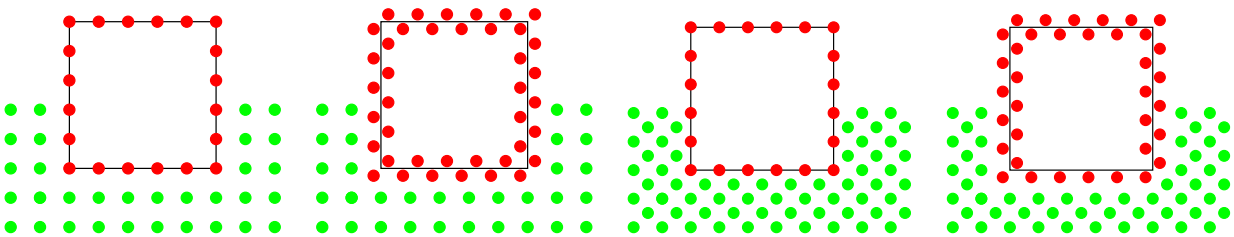
(c) 5 steps

Fig. 6 Effect of the moving least squares filter on the smoothness of the solution. Density contours (kg/m^3) are presented for a cube dropped on the surface of water.



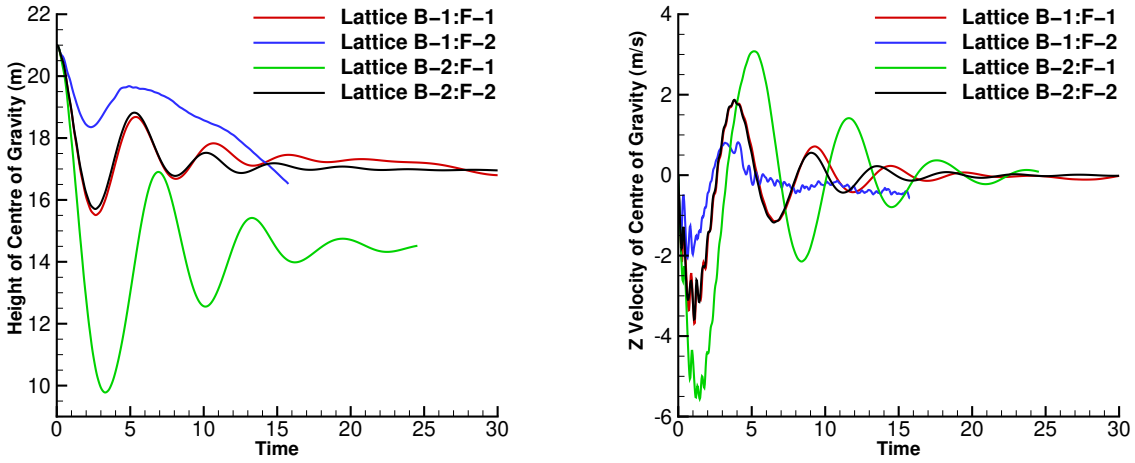
(a) Acceleration (b) Velocity

Fig. 7 Effect of the boundary condition on the Acceleration and velocity of the dropped cube.



(a) Lattice B-1:F-1 (b) Lattice B-2:F-1 (c) Lattice B-1:F-2 (d) Lattice B-2:F-2

Fig. 8 Four combinations of the two different lattices for both the fluid and floating object.



Height above the bottom of the tank

Vertical velocity of the cube

Fig. 9 Effect of the four different lattice combinations on the height and vertical velocity of the centre of gravity of a cube. Lattice B-*x*:F-*y* defines the different combinations where *x* is the lattice type of the boundary and *y* is the lattice type of the fluid. (see figure 8)

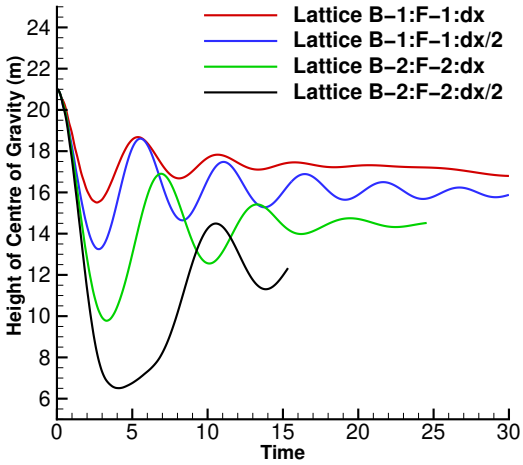
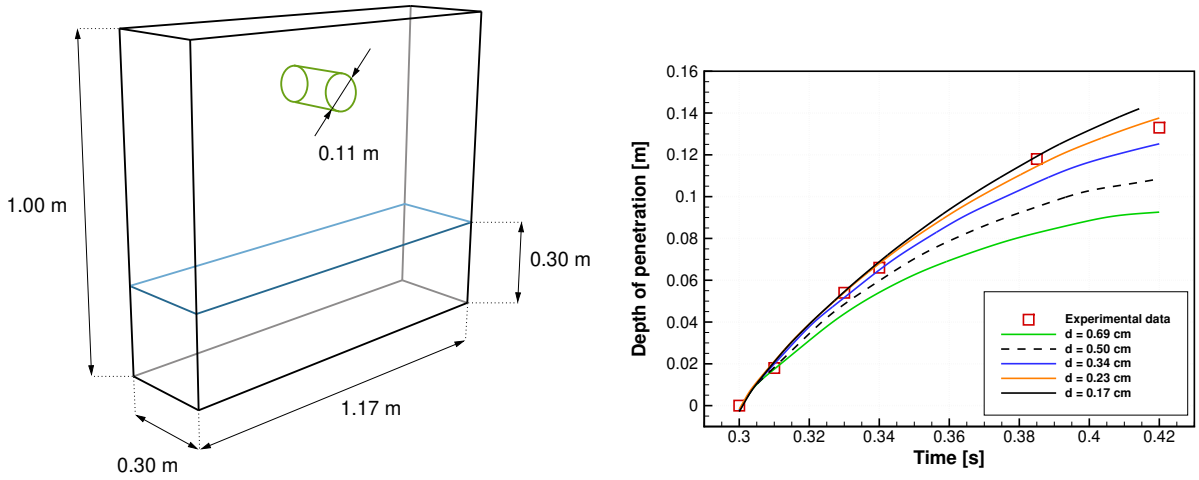


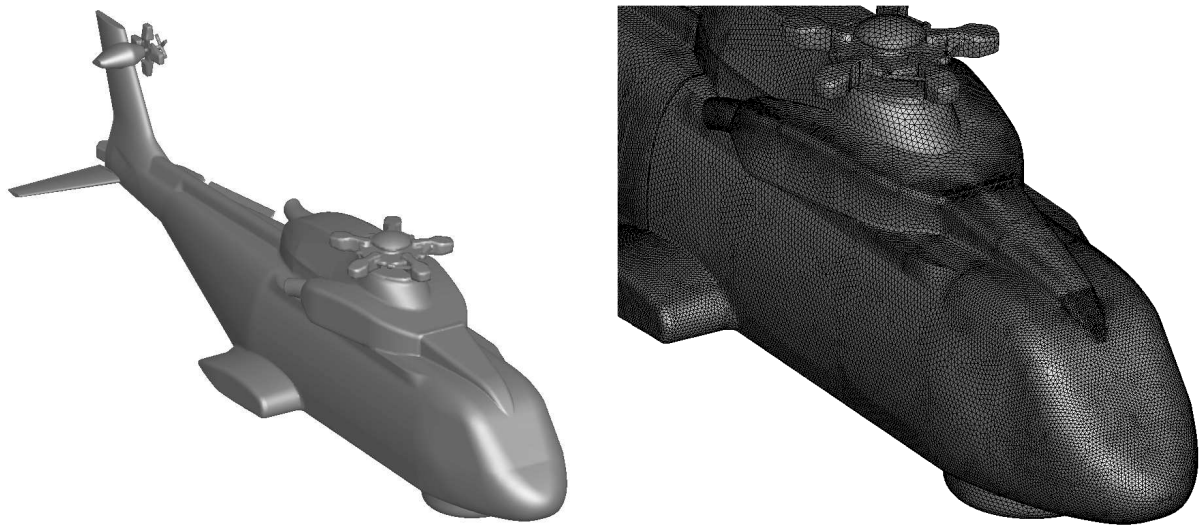
Fig. 10 Effect of increasing the number of particles in the consistent lattice B-1:F-1:dx and lattice B-2:F-2:dx cases where both the boundary and fluid are modelled using type-1 and type-2 lattices respectively, and dx is particle spacing.



(a) Schematic of the SPH validation setup.

(b) Depth of penetration of a cylinder.

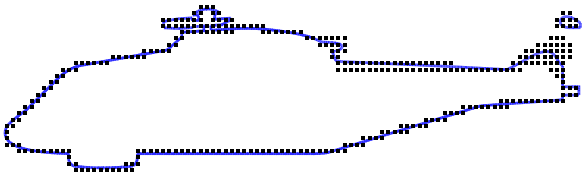
Fig. 11 Validation case for the SPH solver. (a) Schematic of the SPH validation setup; (b) Depth of penetration of a cylinder of density 500 kg/m^3: SPH results for different distances between particles (d) and experimental results of Greenhow and Lin[58]. Simulations were run with a cubic spline kernel, artificial viscosity with viscosity parameter $\alpha = 0.1$, adiabatic index $\gamma = 7$, and Courant-Friedrichs-Lewy number $CFL = 0.2$.



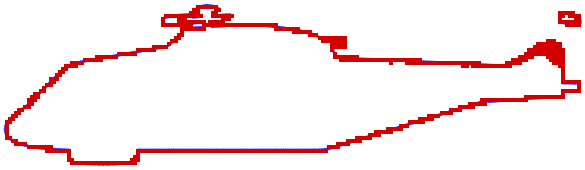
Approximate surface of the AW101 fuselage

Mesh density in STL file

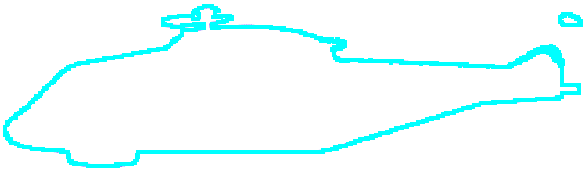
Fig. 12 Surface and mesh of the STereoLithography of the AW101 fuselage shape used in the SPH simulations.



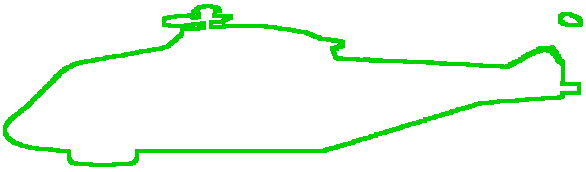
(a) 20 cm particle resolution



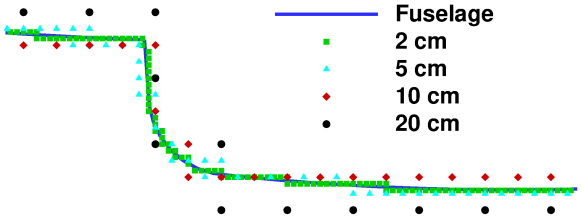
(b) 10 cm particle resolution



(c) 5 cm particle resolution



(d) 2 cm particle resolution



(e) Closeup of the radar dome with all resolutions

Fig. 13 Effect of particle resolution on a section through the fuselage.

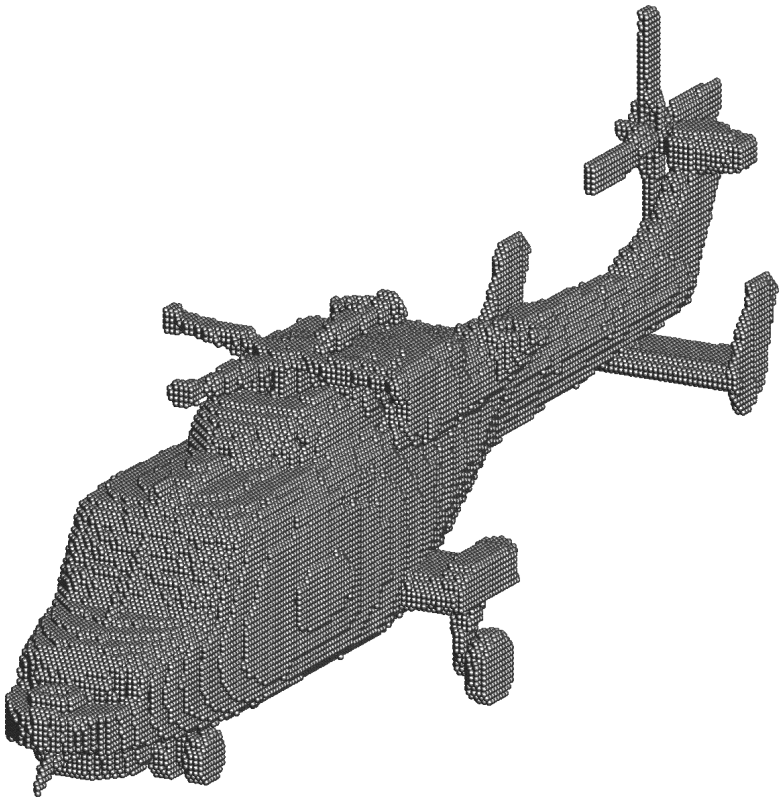
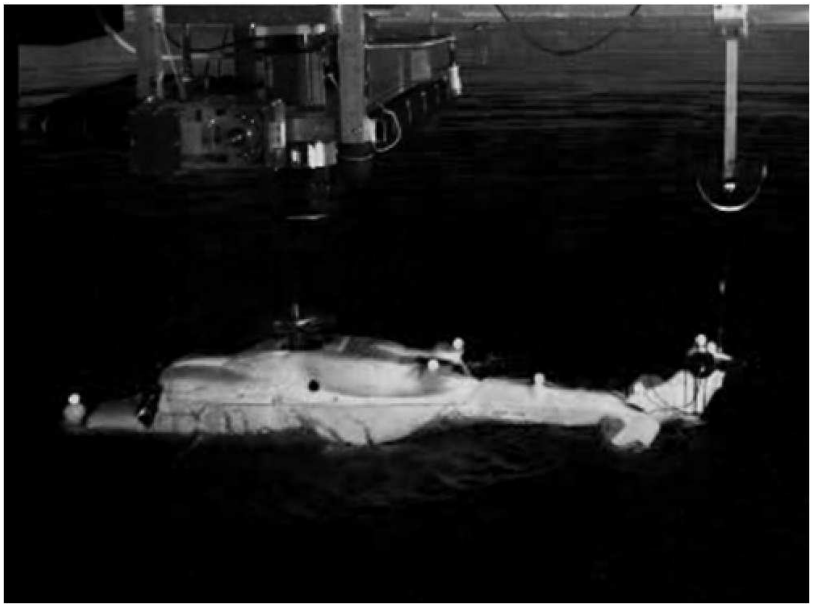


Fig. 14 Particles used to represent the AW159 fuselage (5cm full scale, or 0.5cm for 1/10 model scale).

(a) The Experimental model



(b) Still of drop test at time 0.2 seconds

Fig. 15 Vertical drop of the AW159 Fuselage with 69% **lift into sea state zero.**

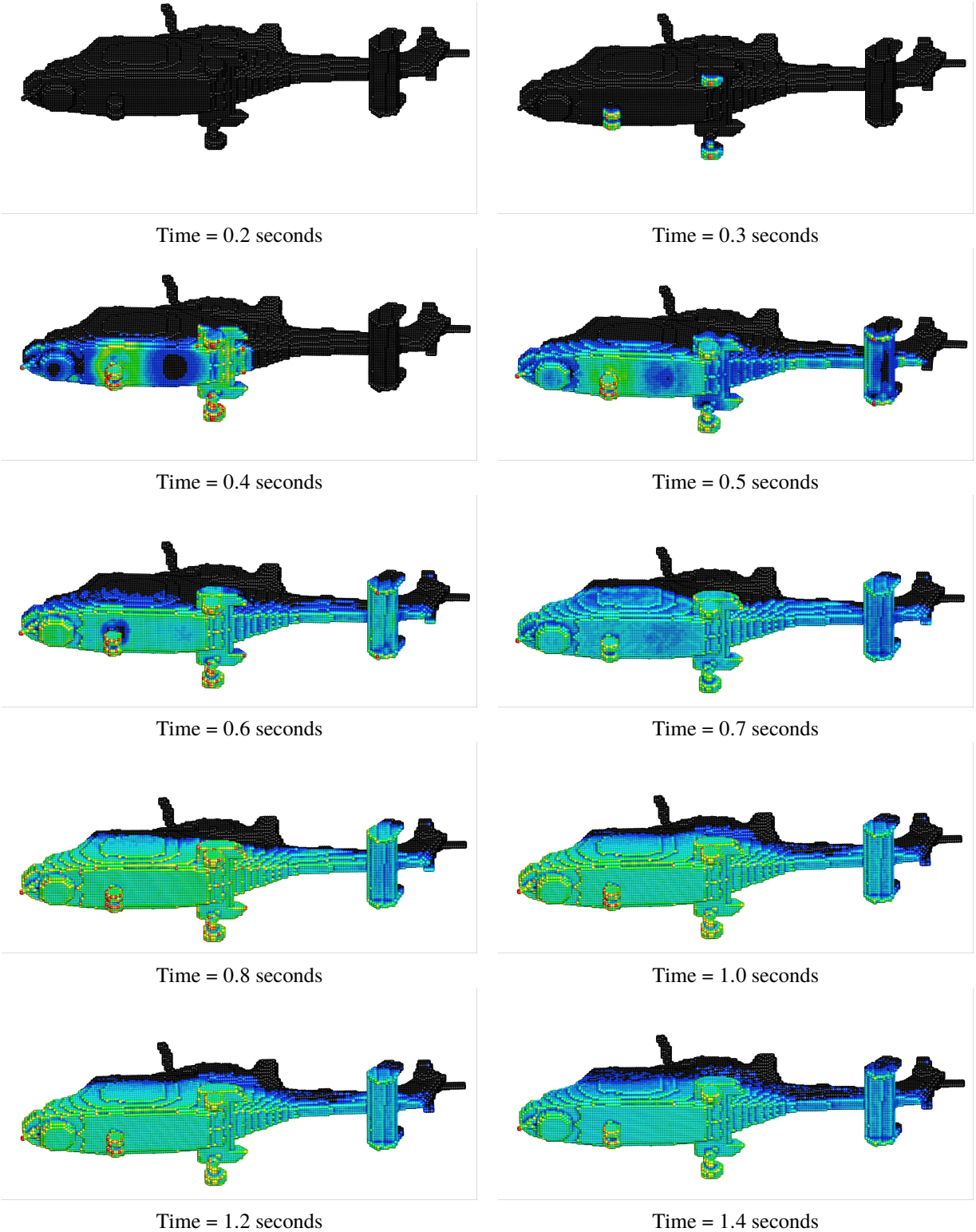
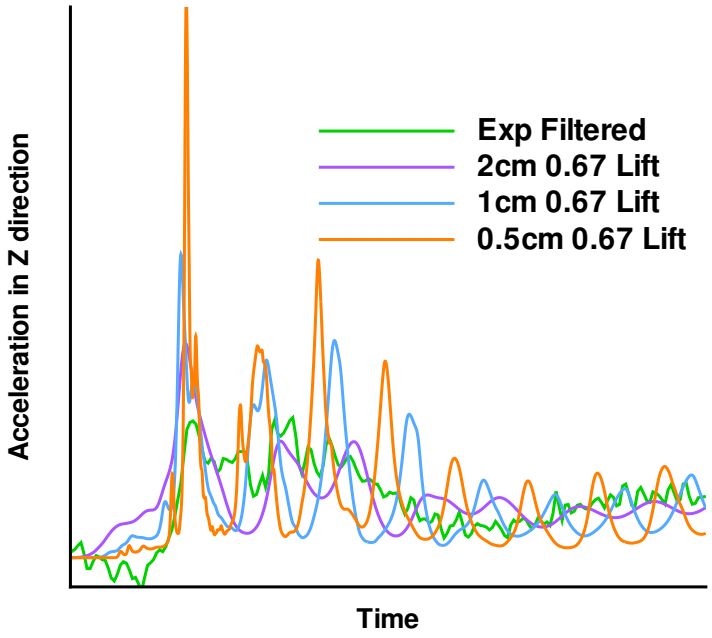
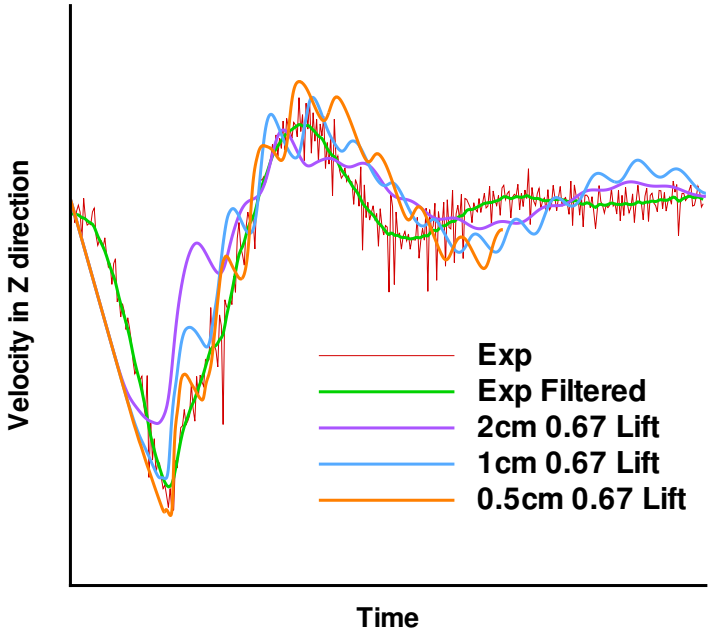


Fig. 16 Pressure on the underside of the AW159 fuselage with 67% lift from the basic rotor model.

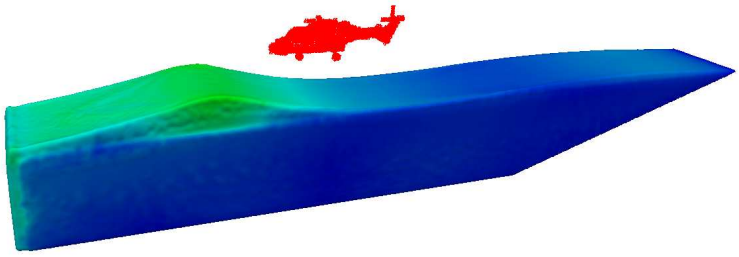


(a) Velocity in the vertical direction

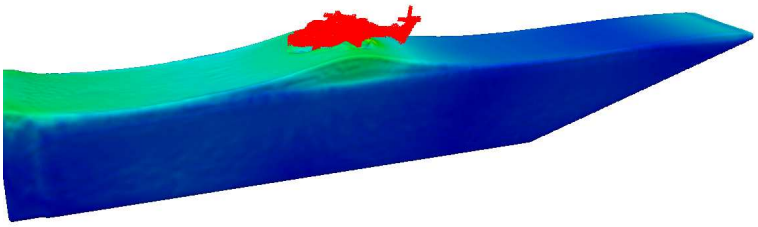


((b) Acceleration in the vertical direction

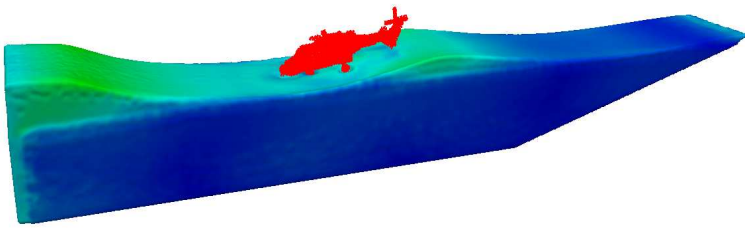
Fig. 17 Comparison between SPH and experimental data for a vertical AW159 Fuselage drop into sea state 0.



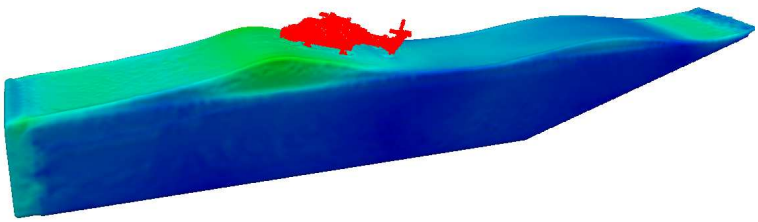
Time = 2.7 Seconds



Time = 3.2 Seconds

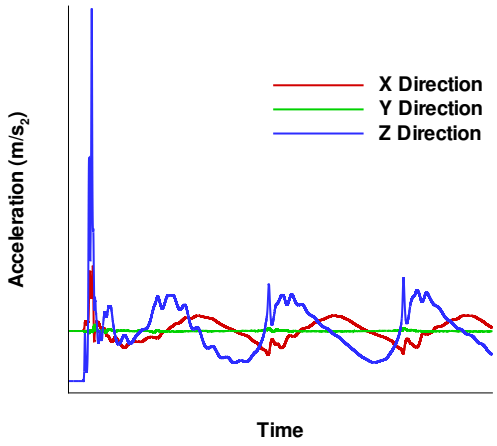


Time = 3.7 Seconds

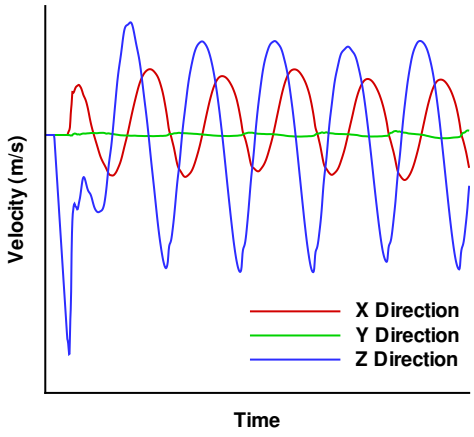


Time = 4.5 Seconds

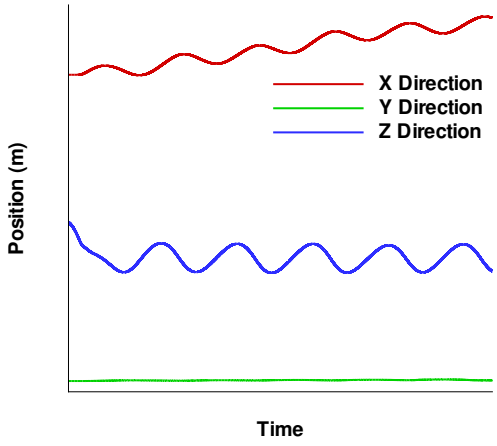
Fig. 18 A vertical drop into sea state 4 and different times with the fuselage hitting the crest of the wave.



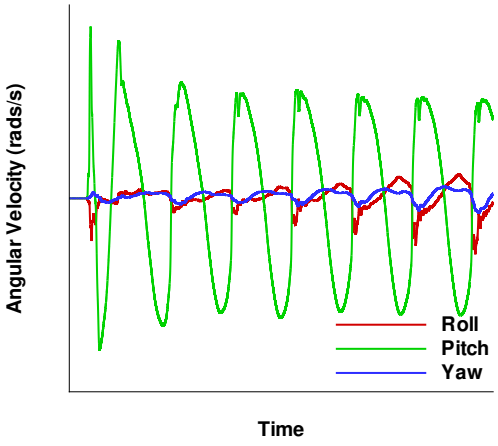
Acceleration of Centre of Gravity



Velocity of Centre of Gravity



Position of Centre of Gravity



Angular velocity of Centre of Gravity

Fig. 19 Vertical drop on sea state 4 **and different times with the fuselage hitting the crest of the wave.**